REMARKS

Claims 1-16 are thought to be allowable over the cited prior art, and the application is thought to be in condition for allowance. Reconsideration and allowance are respectfully requested.

The Office Action fails to show that claims 1-16 are anticipated under 35 USC §102(b) by "Davidson" (US patent 5,613,117 to Davidson et al.). The rejection is respectfully traversed because the Office Action fails to show that all the limitations of the claims are taught by Davidson.

Claim 1 sets forth a method for switching between multiple implementations of a routine in a library of routines that are linked with an application program that is hosted by a computer system, and claim 16 sets forth a computer program product configured to perform the steps of claim 1. The limitations include compiling a plurality of implementations of a routine into respective object code modules, the routine having an associated name and each implementation adapted to a selected hardware configuration; associating the object code modules with the name of the routine and respective sets of hardware characteristics; and resolving when the application program is loaded into memory of the computer system, a reference to the routine using the sets of hardware characteristics and a hardware configuration of the system.

Among other limitations not shown to be taught by Davidson, the Office Action clearly does not show that Davidson teaches associating the object code modules with the name of the routine and respective sets of hardware characteristics. For example, the Office Action cites Davidson's 9:8-15 as teaching these limitations. However, there are no apparent associations of names with sets of hardware characteristics. The cited section of Davidson teaches:

> To create an object module 23, the front end 20 translates the input stream or some subsequence thereof (which we can call a source module 21) into its internal representation for interface 22, which consists of a symbol table 30 for the module and an intermediate language graph 55 for each routine. The front end 20 then calls back end routines to initialize the object module 23, to allocate storage for the symbols in the symbol table 30 via storage allocation 28,

to initialize that storage, to generate code for the routines via emitter 29, and to complete the object module 23. (9:8-18).

There no apparent suggestion of any such claimed association, nor is there any apparent suggestion of any respective sets of hardware characteristics. In view of the apparent lack of corresponding elements, an explanation is requested if the rejection is maintained. Otherwise, the rejection should be withdrawn.

The Office Action also clearly does not show that Davidson teaches resolving when the application program is loaded into memory of the computer system, a reference to the routine using the sets of hardware characteristics and a hardware configuration of the system. The Office Action cites Davidson's 27:23-32. However, there are no apparent elements of Davidson that correspond to these claim limitations as the cited text clearly demonstrates. The cited text reads:

> The design of the compiler of FIG. 1 in general, and of the intermediate language and symbol table in particular, is intended to address a variety of architectures ranging from "Complex Instruction Set Computers" (CISC) such as VAX to "Reduced Instruction Set Computers" (RISC) such as PRISM, MIPS (a 32-bit RISC machine), or an advanced 64-bit RISC architecture. This design does assume that the architecture of target machine 25 has certain basic features. First byte organization and addressability are assumed and Twos-complement binary arithmetic, with "Little-endian" bit ordering. "Reasonable" address representation is also assumed, i.e., that an address fits in a register.

There is no apparent resolving of a reference to a routine, as clearly claimed. Nor is there any apparent resolving performed when a program is loaded. Furthermore, Davidson teaches that the byte organization and addressability features are assumed. By assuming these features, the implication is that Davidson would not need to perform any resolution based on hardware characteristics when a program is loaded.

In failing to establish that Davidson teaches all the limitations of claims 1 and 16, the Office Action fails to show that claims 1 and 16 are anticipated by Davidson.

Claim 2 depends from claim 1 and is not shown to be anticipated for at least the reasons set forth above.

Claim 3 includes further limitations that, for the routine having a plurality of implementations, a plurality of entries are added to the symbol table and respective sets of hardware characteristics are associated with the plurality of entries. It is

respectfully submitted that the Office Action fails to show any association with sets of hardware characteristics. The Office Action cites Davidson's symbol table and Davidson's assumed basic features of a hardware architecture (byte organization, addressability, twos-complement arithmetic, and Little-endian). However, as explained above, Davidson's hardware assumptions obviate the need for any association as claimed. Thus, claim 3 is not shown to be anticipated.

Claim 4 depends from claim 3 and is not shown to be anticipated for at least the reasons set forth above.

Claim 5 depends from claim 4, and claim 6 depends from claim 3. These claims include the further limitations of the resolving step including obtaining the hardware configuration of the system from at least one of a system configuration data file, one or more system identification registers, and system firmware. As explained above in regards to claim 1, Davidson is not shown to perform any resolving. Thus, the further limitations related to the resolving are also not shown to be taught by Davidson. Furthermore, the cited teaching of Davidson (6:43-47) simply indicates that the target machine for Davidson's backend is a computer with a specific architecture. There is no apparent suggestion of obtaining the information from the specifically claimed sources. An explanation of Davidson's elements thought to correspond to the claim limitations is requested if the rejection is maintained. Otherwise, the Office Action fails to show that Davidson anticipates claims 5 and 6, and the rejection should be withdrawn.

Claim 7 depends from claim 1 and is not shown to be anticipated for at least the reasons set forth above.

Claim 8 includes limitations similar to those of claims 5 and 6 and discussed above. Thus, claim 8 is not shown to be anticipated.

Claim 9 includes limitations that further refine the limitations of claim 1 and is not shown to be anticipated by Davidson for at least those reasons set forth above. In addition, claim 9 includes limitations of having a set of hardware characteristics in a symbol table. The cited section of Davidson contains no apparent teaching of hardware characteristics in a symbol table, even though a symbol table is mentioned. Furthermore, Davidson contains no apparent teaching of the claimed matching using the set of hardware characteristics. An explanation of Davidson's elements thought to

correspond to these claim limitations is requested if the rejection is maintained. Otherwise, the Office Action fails to show that Davidson anticipates claim 9, and the rejection should be withdrawn.

Claim 10 includes limitations similar to those of claim 4, and the Office Action does not show that claim 10 is anticipated for at least the reasons set forth above.

Claims 11 and 12 include limitations similar to those of claims 5 and 6, and the Office Action does not show that claims 11 and 12 are anticipated for at least the reasons set forth above.

Claim 13 is an apparatus claim that includes limitations similar to those of claim 1. Thus, claim 13 is not shown to be anticipated by Davidson.

Claim 14 sets forth a computer-implemented symbol table for referencing a library of object code modules that implement a plurality of routines. The limitations include a first set of one or more entries, each entry in the first set including a unique name of a routine and a reference to an object code module in the library; and a second set of one or more entries, each entry in the second set including a shared name of a routine, a set of hardware characteristics, and a reference to an object code module in the library. It is respectfully submitted that the Office Action does not show that Donaldson teaches a symbol table having a set of hardware characteristics. The cited teaching at 29:1-5 references a "literal table." However, there is no apparent correspondence of an element in Davidson to the limitations of a set of hardware characteristics in a symbol table. An explanation of a correspondence is requested if the rejection is maintained. Otherwise, the Office Action has not shown that Davidson anticipates claim 14.

Claim 15 depends from claim 14, and is not shown to be anticipated for at least the reasons set forth above.

The Office Action does not establish that claims 1-16 are anticipated by Davidson. Further clarification is requested if the rejection is maintained. Otherwise the rejection should be withdrawn.

Withdrawal of the rejection and reconsideration of the claims are respectfully requested. No extension of time is believed to be necessary for consideration of this response. However, if an extension of time is required, please consider this a petition

for a sufficient number of months for consideration of this response. If there are any additional fees in connection with this response, please charge Deposit Account No. 50-0996 (HPCO.008PA).

Respectfully submitted,

CRAWFORD MAUNU PLLC
1270 Northland Drive, Suite 390
Saint Paul, MN 55120
(651) 686-6633

By: _____

Name: LeRoy D. Maunu
Reg. No.: 35,274